LA-UR-22-20381

Approved for public release; distribution is unlimited.

Title:	Agile Project Management for Research Software
Author(s):	Herring, Angela M.
Intended for:	Internal seminar
Issued:	2022-01-18

Los Alamos
NATIONAL LABORATORY

Los Alamos NATIONAL LABORATORY

Agile Project Management for Research Software

Angela Herring

January 18, 2022

Why should scientists like Scrum?

- We like to control our own destiny!
  - Scrum allows the team to stay in control of research and process
- Research is unpredictable
  - Scrum helps us manage uncertainty
- Collaboration is critical to multi-disciplinary projects
  - Allows team members to cycle in and out as expertise needed changes
- Scrum provides a lean and robust reporting/planning framework
  - Add tickets to make your monthly highlights!
- Process serves the team
  - Change it when you need to!

Scrum allows research teams to stay organized, manage change, enable necessary reporting and collaborate with minimal effort.

Scrum in 100 words

- Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.
- It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).
- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.
- Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.

Scrum has been used by:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- IBM
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce

Scrum has been used for:

- Commercial software
- In-house development
- Contract development
- Fixed-price projects
- Financial applications
- ISO 9001-certified applications
- Embedded systems
- 24x7 systems with 99.999% uptime requirements
- the Joint Strike Fighter
- Video game development
- FDA-approved, life-critical systems
- Satellite-control software
- Websites
- Handheld software
- Mobile phones
- Network switching applications
- ISV applications
- Some of the largest applications in use

Characteristics

- Self-organizing teams
- Product progresses in a series of 1–4 week "sprints"
- Requirements are captured as items in a list of "product backlog"
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects
- One of the "agile processes"

Process overview

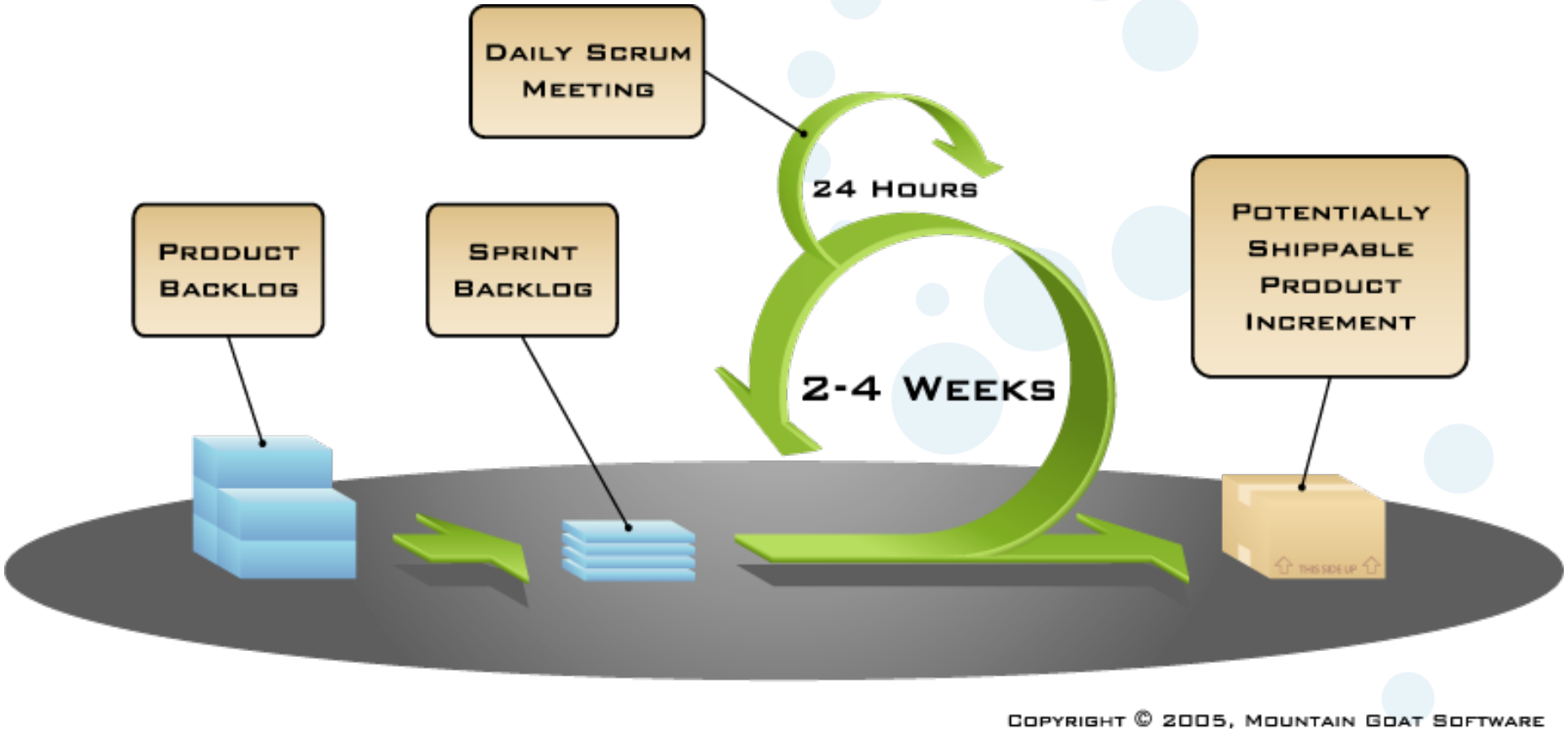
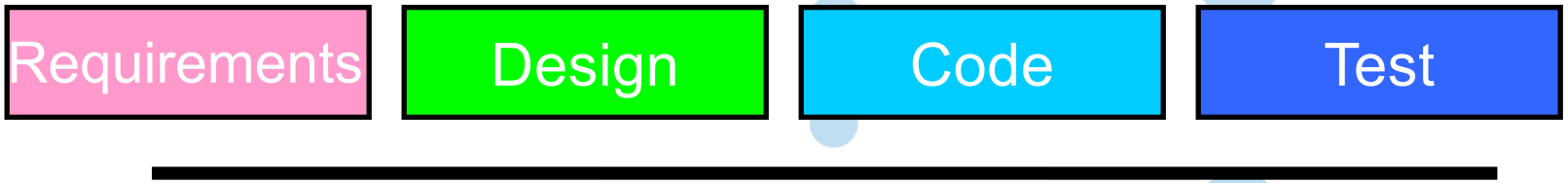


Image available at
www.mountaingoatsoftware.com/scrum

Sprints

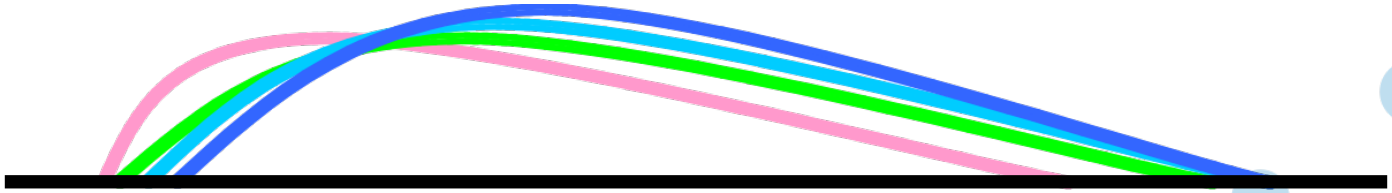
- Scrum projects make progress in a series of “sprints”
- Analogous to Extreme Programming iterations
- Typical duration is 2–4 weeks or a calendar month at most
- A constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint

Sequential vs. overlapping development



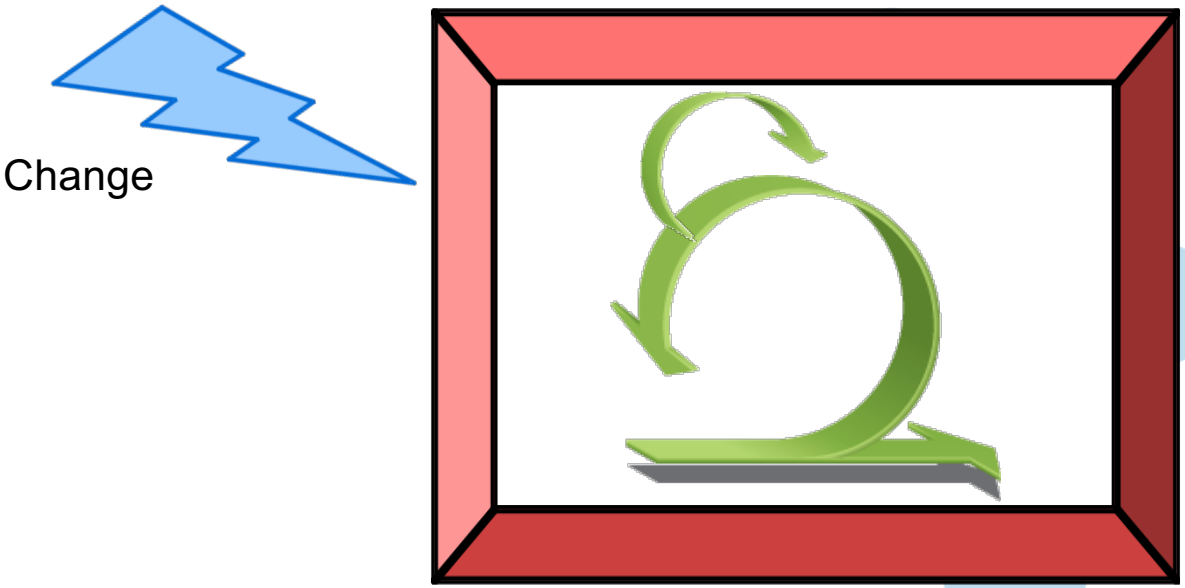
Rather than doing all of one thing at a time...

...Scrum teams do a little of everything all the time



Source: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review,* January 1986.

No changes during a sprint



Plan sprint durations around how long you can commit to keeping change out of the sprint

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

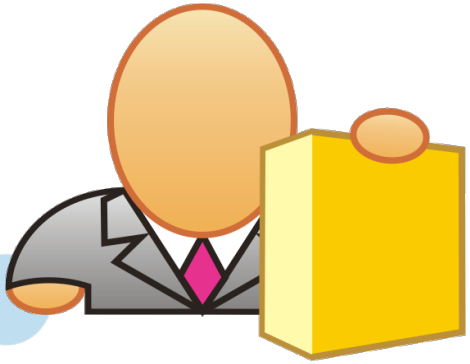
Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product owner

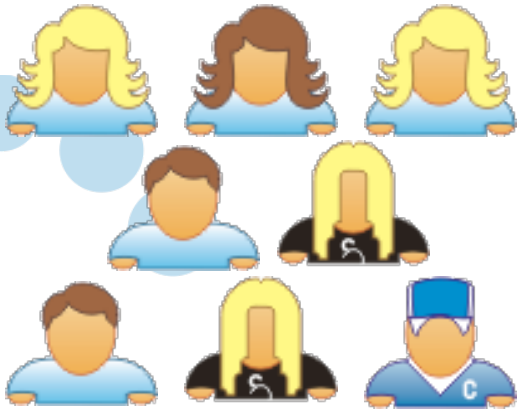


- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results

The ScrumMaster

- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences

The team



- Typically 5–9 people
- Cross-functional:
  - Programmers, testers, user experience designers, etc.
- Members should be full-time
  - May be exceptions (e.g., database administrator)
- Teams are self-organizing
  - Ideally, no titles but rarely a possibility
- Membership should change only between sprints

Scrum framework

Roles

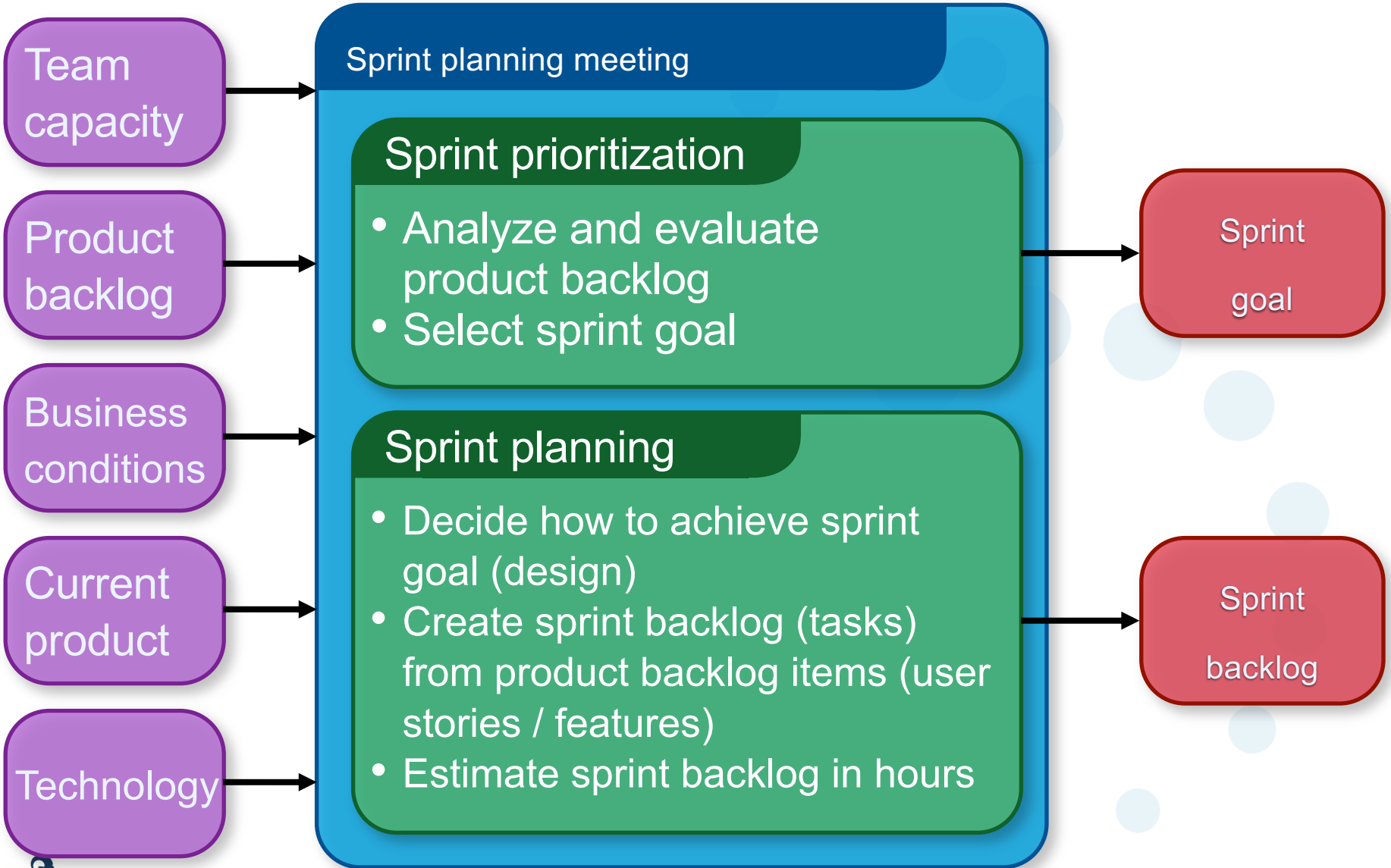
- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts



Sprint planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
- Tasks are identified and each is estimated (1–16 hours)
- Collaboratively, not done alone by the ScrumMaster
- High-level design is considered

As a vacation planner, I want to see photos of the hotels.

Code the middle tier (8 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product backlog

This is the product backlog

- The requirements
- A list of all desired work on the project
- Ideally expressed such that each item has value to the users or customers of the product
- Prioritized by the product owner
- Reprioritized at the start of each sprint

A sample product backlog

Backlog item	Estimate
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

The sprint goal

A short statement of what the work will be focused on during the sprint

Life Sciences

Support features necessary for population genetics studies.

Database Application

Make the application run on SQL Server in addition to Oracle.

Financial services

Support more technical indicators than company ABC with real-time, streaming data.

Managing the sprint backlog

- Individuals sign up for work of their own choosing
  - Work is never assigned
- Estimated work remaining is updated daily
- Any team member can add, delete or change the sprint backlog
- Work for the sprint emerges
- If work is unclear, define a sprint backlog item with a larger amount of time and break it down later
- Update work remaining as more becomes known

A sprint backlog

Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

Scrum framework

Roles

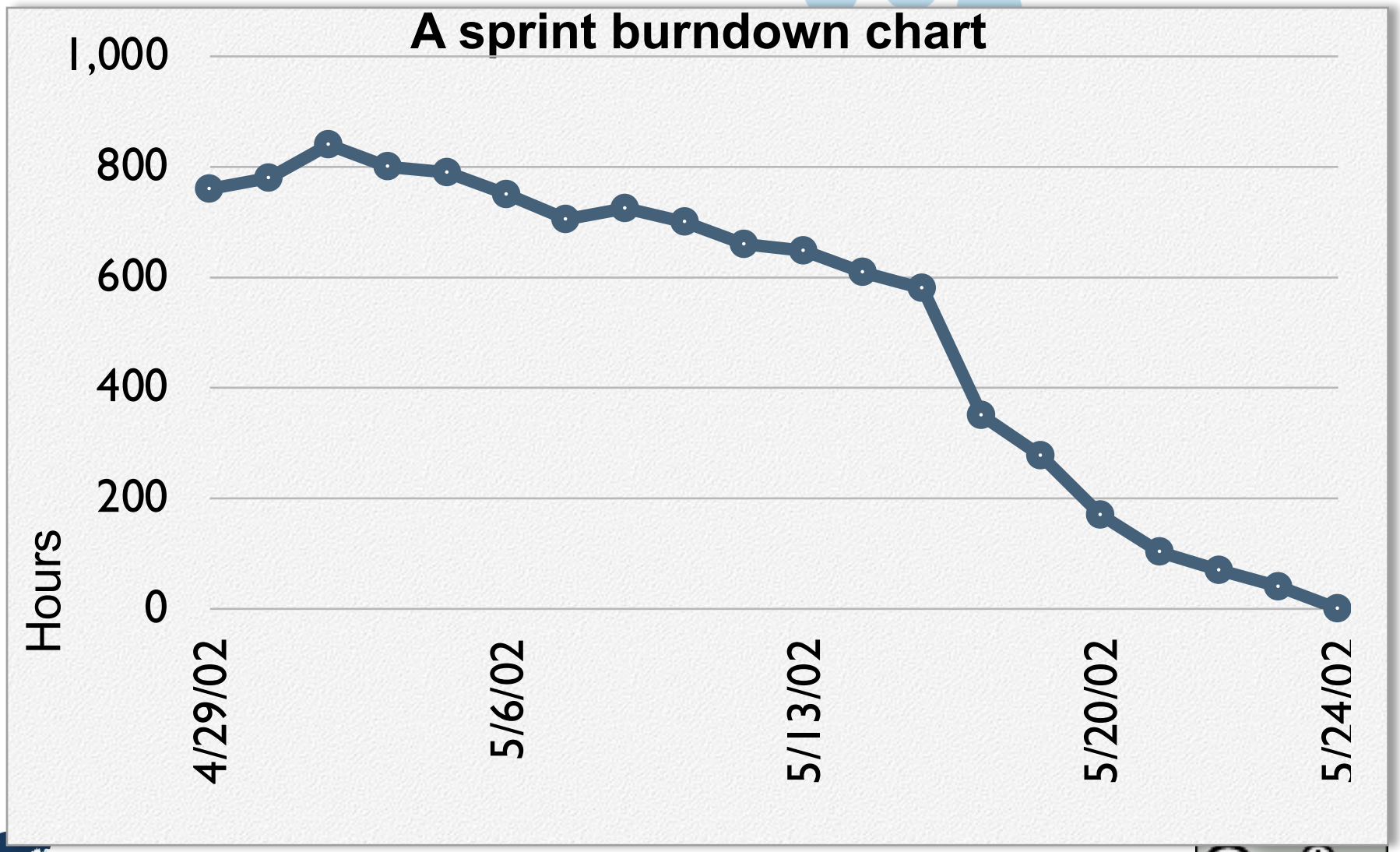
- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts


A sprint burndown chart
1,000
800
600
400
200
0
Hours
4/29/02
5/6/02
5/13/02
5/20/02
5/24/02

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

The daily scrum

Parameters:

- Daily
  - 15 minutes
  - Stand-up
- Not for problem solving
- Whole world is invited
- Only team members, ScrumMaster, product owner, can talk
- Helps avoid other unnecessary meetings

Everyone answers 3 questions

1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

These are *not* status for the ScrumMaster

They are commitments in front of peers

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

The sprint review

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
  - 2-hour prep time rule
  - No slides
- Whole team participates
- Invite the world

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Sprint retrospective

- Periodically take a look at what is and is not working
- Typically 15–30 minutes
- Done after every sprint
- Whole team participates
  - ScrumMaster
  - Product owner
  - Team
  - Possibly customers and others

Start / Stop / Continue

Whole team gathers and discusses what they'd like to:

Start doing

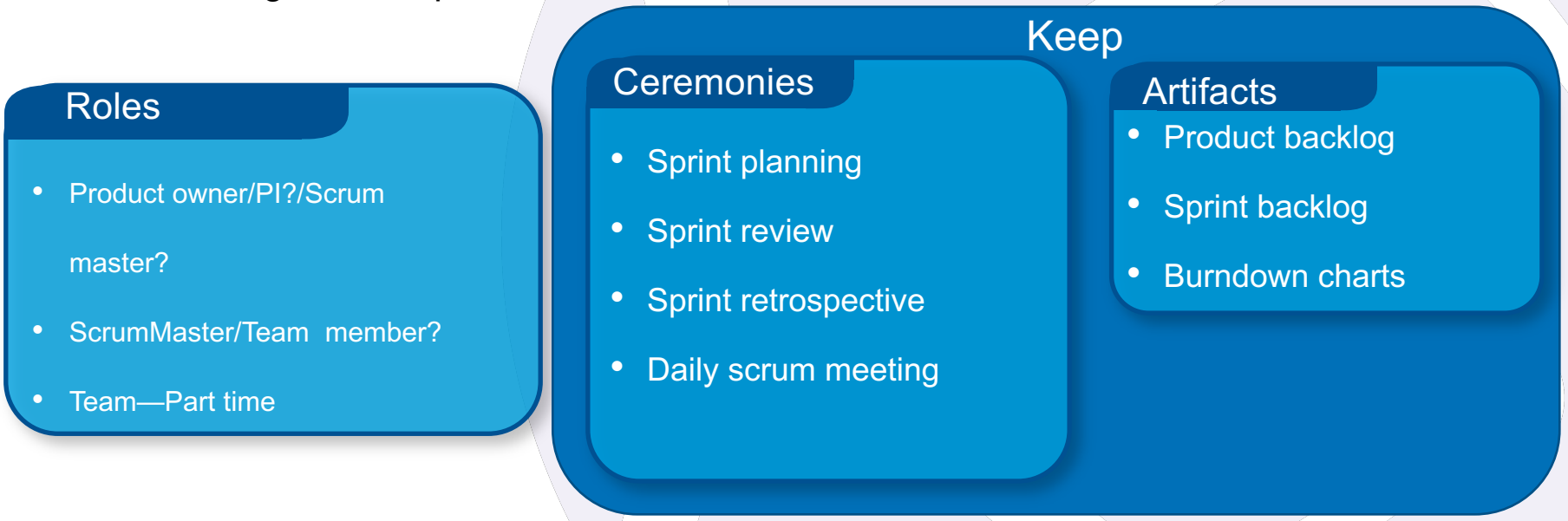
Stop doing

Continue doing

This is just one of many ways to do a sprint retrospective.

Rules were made to be broken or: How I learned to stop worrying and love the sprint

- Changes to the scrum process are required for LANL environment:
  - Part-time staff
  - Dual-hatted roles, allowing scrum master to be team member or product owner
  - Less rigorous tracking of per-task labor
  - <20 minute prep time for sprint demo
  - Team sets goals and priorities!!



Keep the ceremonies and artifacts; muddy the waters with roles as needed.

Agile principles: Stick to the code

- Self-organizing, collaborative teams are the key to the whole process:
  - Team has a flat hierarchy
  - Team owns the scrum process and can change it
  - Work is never assigned
  - No one “owns” code or tasks
  - Encourage staff to take a task they don’t know how to do each sprint
  - Watch out for each other
    - Review tasking, did someone take too much?
    - How can you help someone who is struggling?
  - Appreciate one another

- Maintain Agile principles:
  - Plan your work in sprints
  - Don’t allow changes to work scope during sprints
  - Have a stand-up to monitor progress and report issues
  - Have a sprint review so folks can show off their work
  - Have a retrospective to enable process changes and problem solving
  - Be open to change

Building a collaborative, resilient, flexible team is the most important thing.

End your sprint planning with a laugh: Name your sprint

- Determine a sprint naming scheme:
  - Try alphabetical with a fun naming scheme:
    - AA, BB, CC …

      Or
    - AB, BC, CD …
- Example naming schemes:
  - Fantasy+Sci Fi (Tolkien Tribbles, Nearly headless Nick Fury, Draco's Dalek)
  - Adjective + animal (Vindictive Weevil, Bereaved Cockroach, Kooky Locust)
- Everyone contributes names, and everyone votes

Bonus points if you can get management to say this out loud at a meeting!

During the sprint: Weekly teatime encourages collaboration and team-building

Parameters

- Set time to get together
- Supply snacks tea/coffee
- Freeform meeting
- Topics of conversation
  - Kids, pets, hobbies
  - Open pull requests
  - Open research topics

Benefits

- Team cohesion
- Team bonding
- Team collaboration
- Easy time to get questions answered, find teammates

Don't be tempted to structure this meeting. Let it evolve around the team's moods/issues/needs.

Retrospectives: those who do not learn from history are doomed to repeat it

- Tempting to skip a retrospective
- Don’t.
- Because they help you
  - Catch problems before they grow
  - Modify your scrum process
  - Allow the team to vent
  - Set the tone for your team’s response to frustration

Remember the golden rule: treat others as you would have them treat you

Appreciations: End your retrospective on a high note

- The format is important
- “I appreciate you _____ for your help debugging feature Y.”
- No one gets out of giving an appreciation
- Appreciations can be simple and should be sincere

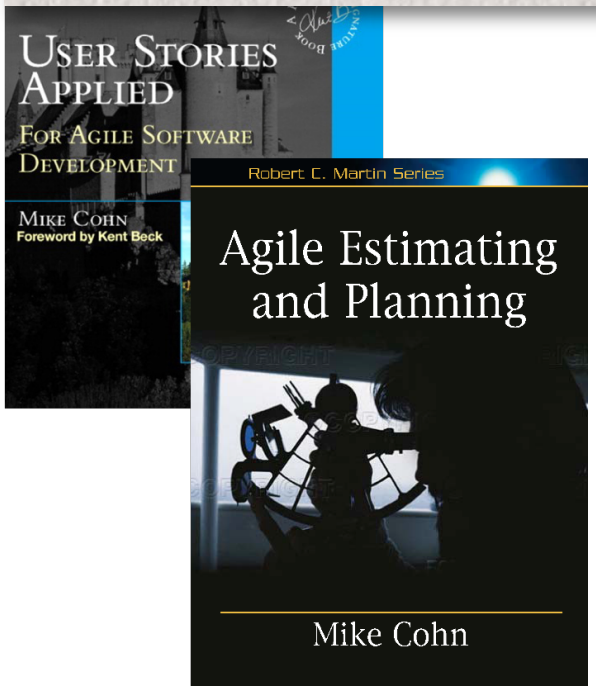
Appreciations remind us that we are all in this together!

Discussion

Overview Presentation by: Mike Cohn

mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com. (720) 890-6110



- Further reading:
  - Herring, A. "A lightweight process for Agile software management." LA-UR-19-25789
  - Herring, A. "Listen, don't just hear your team with an effective retrospective." LA-UR-21-21952
  - Herring, A. "BSSW Thanksgiving blog post." LA-UR-19-31693
  - Beedle, Schwaber. Agile software development with Scrum. ISBN 978-0130676344
  - Derby, Larsen. Agile Retrospectives: Making Good Teams Great. ISBN-13: 978-0977616640